ABSTRACT

A split-stream dictionary (SSD) program compression architecture has a dictionary builder, a dictionary compressor, and a SSD item generator. The dictionary builder constructs a dictionary containing two types of entries: (1) base entries for each instruction in an input program and (2) sequence entries for sequences of multiple instructions (e.g., 2-4 instructions) that are used multiple times in the program. The dictionary compressor compresses the dictionary by separately compressing the base entries and the sequence entries independently of one another. The SSD item generator generates a stream of items that represent the program instructions in terms of the base entries and the sequence entries. The SSD program compression architecture outputs the compressed dictionary and the stream of SSD items referencing the dictionary. The SSD program compression architecture supports a two-phase just-in-time (JIT) translation having a dictionary decompression phase and a copy phase. In the decompression phase, the VM loads and decompresses the dictionary. In the copy phase, the VM expands each basic block by copying dictionary entries into a native code buffer, thereby effectively translating the SSD items back into the instructions.